

PI
1.

PPPPPPPP	LL		PPPPPPPP	UU	UU	TTTTTTTTTT	LL		SSSSSS
PPPPPPPP	LL		PPPPPPPP	UU	UU	TTTTTTTTTT	LL		SSSSSS
PP	PP		PP	PP	UU	TT	LL		SS
PP	PP		PP	PP	UU	TT	LL		SS
PP	PP		PP	PP	UU	TT	LL		SS
PP	PP		PP	PP	UU	TT	LL		SS
PPPPPPPP	LL		PPPPPPPP	UU	UU	TT	LL		SSSSSS
PPPPPPPP	LL		PPPPPPPP	UU	UU	TT	LL		SSSSSS
PP	LL		PP	UU	UU	TT	LL		SS
PP	LL		PP	UU	UU	TT	LL		SS
PP	LL		PP	UU	UU	TT	LL		SS
PP	LL		PP	UU	UU	TT	LL		SS
PP	LLLLLLLL		PP	UUUUUUUU	TT	LLLLLLLL		SSSSSS
PP	LLLLLLLL		PP	UUUUUUUU	TT	LLLLLLLL		SSSSSS
LL		SSSSSS							
LL		SSSSSS							
LL		SS							
LL		SS							
LL		SSSSSS							
LL		SSSSSS							
LL		SS							
LL		SS							
LL		SS							
LL		SS							
LLLLLLLL		SSSSSS							
LLLLLLLL		SSSSSS							

```
0000 1 .title plisputlistitem
0000 2 :ident /1-002/ ; Edit WHM1002
0000 3
0000 4
0000 5 ****
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 *
0000 25 *
0000 26 ****
0000 27 :
0000 28
0000 29 ++
0000 30 :facility:
0000 31 :      VAX/VMS PL1 runtime library
0000 32
0000 33
0000 34 :abstract:
0000 35
0000 36 :      This module contains the pl1 runtime routines to put items to a
0000 37 :      pl1 stream file under list directed i/o.
0000 38
0000 39
0000 40 :author: c. spitz 28-nov-79
0000 41
0000 42 :modified:
0000 43
0000 44
0000 45 :      1-002 Bill Matthews 29-September-1982
0000 46
0000 47 :      Invoke macros $defdat and rtshare instead of $defopr and share.
0000 48
0000 49 :--
0000 50
0000 51 :external definitions
0000 52
0000 53 :
0000 54 :      $deffcbl :define file control block
0000 55 :      $defstk :define stack frame offsets
0000 56 :      $defstr :define stream block offsets
0000 57 :      $defdat :define operand node data types
```

```

0000 58 $def getopt :define get options block
0000 59 $rabdef :define rms rab offsets
0000 60 $rmsdef :define rms error codes
0000 61 $ssdef :define system status codes
0000 62
0000 63 :
0000 64 : local data
0000 65 :
0000 66
0000 67 rtshare ;sharable
0000 68
0000 69 :++
0000 70 pli$putl***+
0000 71
0000 72 : the pli$putl*** routines are called by the compiled code to put items
0000 73 : to a stream output file under list directed transmission. each routine
0000 74 : converts the source item to a character string based on the source data
0000 75 : type, and puts then puts the string to the file by jumping to
0000 76 : pli$putnlis_r6.
0000 77 :-
0000 78
0000 79
0000 80 pli$putlchar_r6
0000 81 : inputs:
0000 82 :   r0 - address of element to put
0000 83 :   r1 - size/prec of element to put
0000 84 :   r11 - address of stream block
0000 85 :   ap - address of file control block
0000 86 : outputs:
0000 87 :   none
0000 88 : side effects:
0000 89 :   r0-r6 are destroyed
0000 90
0000 91 pli$putlchar_r6:::
0000 92 bisl #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
0000 93 addl r0,r1 ;get ending address of source
0000 94 movab <str_b_field+2>(r11),r2 ;get starting addr in field
0000 95 movl r2,r4 ;copy it
0000 96 movl str_l_fld_end(r11),r3 ;get end addr of field
0000 97 bbs #atr_v_print,fcb_l_attr(ap),5$ ;skip lead quote if print
0000 98 movb #'x27,(r2)+ ;insert the leading quote
0000 99 5$: cmpl r0,r1 ;nothing in source?
0000 100 beql 60$ ;if eql, then yes
0000 101 10$: cmpb (r0),#'x27 ;next char a quote?
0000 102 bneq 20$ ;if neg, then no
0000 103 bbs #atr_v_print,fcb_l_attr(ap),20$ ;if print file, don't change
0000 104 movw #'x2727,(r2)+ ;insert 2 quotes
0000 105 brb 50$ ;continue
0000 106 20$: movb (r0),(r2)+ ;copy to field
0000 107 50$: aoblss r1,r0,55$ ;if not end of source, cont
0000 108 brb 60$ ;finish it off
0000 109 55$: cmpl r2,r3 ;field overflow?
0000 110 blss 10$ ;if lss, then no, cont
0000 111 movl #pli$_strovfl,r0 ;set field overflow
0000 112 brw fail ;and fail
0000 113 60$: bbs #atr_v_print,fcb_l_attr(ap),70$ ;if print, don't add trail quote
0000 114 movb #'x27,(r2)+ ;insert trailing quote

```

```

52 54 C2 0050 115 70$· subl r4,r2 ;get length
74 52 B0 0053 116 movw r2,-(r4) ;set length in field
00000000'GF 16 U056 117 jsb g^pli$putnlis_r6 ;put in buffer
OC AC 08 CA 005C 118 bicl #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
05 0060 119 rsb ;return
0061 120
0061 121:pli$putlvcha_r6
0061 122:inputs:
0061 123:    r0 - address of element to put
0061 124:    r1 - size/prec of element to put
0061 125:    r11 - address of stream block
0061 126:    ap - address of file control block
0061 127:outputs:
0061 128:    none
0061 129:side effects:
0061 130:    r0-r6 are destroyed
0061 131:
0061 132:
0061 133:pli$putlvcha_r6::
0061 134:    bisl #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
0061 135:    movzwl (r0)‡,r1 ;get length of source
0061 136:    brw pli$putlchar_r6 ;continue in common
0068 137
0068 138
0068 139
0068 140:pli$putlbit_r6
0068 141:inputs:
0068 142:    r0 - address of element to put
0068 143:    r1 - size/prec of element to put
0068 144:    r2 - offset to starting bit
0068 145:    r11 - address of stream block
0068 146:    ap - address of file control block
0068 147:outputs:
0068 148:    none
0068 149:side effects:
0068 150:    r0-r6 are destroyed
0068 151:
0068 152:
0068 153:pli$putlbit_r6::
0068 154:    bisl #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
0068 155:    movl r2,r5 ;copy offset
0068 156:    movab str b_field(r11),r2 ;get field addr
0068 157:    addw3 r1,‡3,(r2)+ ;set size
0068 158:    movb #^x27,(r2)+ ;insert a quote
0068 159:    movl r1,r3 ;get size
0068 160:    cmpl r3,#1000 ;field overflow?
0068 161:    bleq 10$ ;if leg, then no
0068 162:    movl #pli$_stroval,r0 ;set field over flow
0068 163:    brw fail ;and fail
0068 164:    addl3 r1,r2,r4 ;get addr of end of string
0068 165:    movw #^x4227,(r4) ;plug in trailing quote and B
0068 166:    calls #0,g^pli$bitchar_r6 ;convert bits
0068 167:    jsb g^pli$putnlis_r6 ;put in buffer
0068 168:    bicl #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
0068 169:    rsb ;return
00AE 170
00AE 171

```

```

00AE 172 :pli$putlbit_r6
00AE 173 :    inputs:
00AE 174 :        r0 - address of element to put
00AE 175 :        r1 - size/prec of element to put
00AE 176 :        r11 - address of stream block
00AE 177 :        ap - address of file control block
00AE 178 :    outputs:
00AE 179 :        none
00AE 180 :    side effects:
00AE 181 :        r0-r6 are destroyed
00AE 182 :
00AE 183 :
00AE 184 pli$putlbit_r6::
00AE 185     bisl  #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
00AE 186     clrl  r2                      ;set offset to 0
00AE 187     brb   pli$putlbit_r6          ;join common code
00AE 188 :
00AE 189 :pli$putlfixb_r6
00AE 190 :    inputs:
00AE 191 :        r0 - address of element to put
00AE 192 :        r1 - size/prec of element to put
00AE 193 :        r11 - address of stream block
00AE 194 :        ap - address of file control block
00AE 195 :    outputs:
00AE 196 :        none
00AE 197 :    side effects:
00AE 198 :        r0-r6 are destroyed
00AE 199 :
00AE 200 :
00AE 201 pli$putlfixb_r6::
00AE 202     bisl  #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
00AE 203     movab str 5,Field(r11),r2      ;set field addr
00AE 204     movl  #1000,r3                  ;set size
00AE 205     calls  #0,g^pli$fixbvcha_r6   ;convert it
00AE 206     jsb   g^pli$putnlis_r6       ;put in buffer
00AE 207     bicl  #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
00AE 208     rsb   r2                      ;return
00AE 209 :
00AE 210 :
00AE 211 :pli$putlfixd_r6
00AE 212 :    inputs:
00AE 213 :        r0 - address of element to put
00AE 214 :        r1 - size/prec of element to put
00AE 215 :        r11 - address of stream block
00AE 216 :        ap - address of file control block
00AE 217 :    outputs:
00AE 218 :        none
00AE 219 :    side effects:
00AE 220 :        r0-r6 are destroyed
00AE 221 :
00AE 222 :
00AE 223 pli$putlfixd_r6::
00AE 224     bisl  #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
00AE 225     movab str 5,Field(r11),r2      ;set field addr
00AE 226     movl  #1000,r3                  ;set size
00AE 227     calls  #0,g^pli$fixdvcha_r6   ;convert it
00AE 228     jsb   g^pli$putnlis_r6       ;put in buffer

```

0C AC 08 CA 00F3 229 bcl #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
05 00F7 230 rsb ;return
00F8 231
00F8 232
00F8 233 :pli\$putlfltb_r6
00F8 234 : inputs:
00F8 235 : r0 - address of element to put
00F8 236 : r1 - size/prec of element to put
00F8 237 : r11 - address of stream block
00F8 238 : ap - address of file control block
00F8 239 : outputs:
00F8 240 : none
00F8 241 : side effects:
00F8 242 : r0-r6 are destroyed
00F8 243 :
00F8 244 :
00F8 245 :pli\$putlfltb_r6::
0C AC 08 C8 00F8 246 bisl #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
52 18 AB 9E 00FC 247 movab str 5_field(r11),r2 ;set field addr
0000003E8 8F D0 0100 248 movl #1000,r3 ;set field width
00000000'GF 00 FB 0107 249 calls #0,g^pli\$fltbvcha_r6 ;convert it
00000000'GF 16 010E 250 jsb g^pli\$putnlis_r6 ;put in buffer
0C AC 08 CA 0114 251 bicl #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
05 0118 252 rsb ;return
0119 253
0119 254
0119 255 :pli\$putlfltd_r6
0119 256 : inputs:
0119 257 : r0 - address of element to put
0119 258 : r1 - size/prec of element to put
0119 259 : r11 - address of stream block
0119 260 : ap - address of file control block
0119 261 : outputs:
0119 262 : none
0119 263 : side effects:
0119 264 : r0-r6 are destroyed
0119 265 :
0119 266
0119 267 :pli\$putlfltd_r6::
0C AC 08 C8 0119 268 bisl #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
52 18 AB 9E 011D 269 movab str 5_field(r11),r2 ;set field addr
0000003E8 8F D0 0121 270 movl #1000,r3 ;set field width
00000000'GF 00 FB 0128 271 calls #0,g^pli\$fltdvcha_r6 ;convert it
00000000'GF 16 012F 272 jsb g^pli\$putnlis_r6 ;put in buffer
0C AC 08 CA 0135 273 bicl #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
05 0139 274 rsb ;return
013A 275
013A 276
013A 277 :pli\$putlpic_r6
013A 278 : inputs:
013A 279 : r0 - address of element to put
013A 280 : r1 - size/prec of element to put
013A 281 : r11 - address of stream block
013A 282 : ap - address of file control block
013A 283 : outputs:
013A 284 : none
013A 285 : side effects:

013A 286 : r0-r6 are destroyed
013A 287
013A 288 pli\$putlpic_r6:
00000000'GF 00 000003E8 8F 00 0142 289 bisl #atr_m_recur,fcb_l_attr(ap) ;set recursion flag
00000000'GF 03 00000000'8F 03 015B 290 movab str b_field(r11),r2 ;set field addr
00000000'GF 03 00000000'8F 03 015B 291 movl #1000,r3 ;set size
00000000'GF 03 00000000'8F 03 015B 292 calls #0,g^pli\$picvcha_r6 ;convert it
00000000'GF 03 00000000'8F 03 015B 293 jsb g^pli\$putnlis_r6 ;put in buffer
00000000'GF 03 00000000'8F 03 015B 294 bicl #atr_m_recur,fcb_l_attr(ap) ;clr recursion flag
00000000'GF 03 00000000'8F 03 015B 295 rsb ;return
00000000'GF 03 00000000'8F 03 015B 296
00000000'GF 03 00000000'8F 03 015B 297 fail: movl r0,fcb_l_error(ap) ;set error in fcb
00000000'GF 03 00000000'8F 03 015B 298 pushl ap ;set fcb address
00000000'GF 03 00000000'8F 03 015B 299 pushl r0 ;set error code
00000000'GF 03 00000000'8F 03 0161 300 pushl #pli\$_error ;set error condition
00000000'GF 03 00000000'8F 03 0163 301 calls #3,g^pli\$io_error ;signal error condition
00000000'GF 03 00000000'8F 03 0169 302 ret ;return
00000000'GF 03 00000000'8F 03 0170 303
00000000'GF 03 00000000'8F 03 0171 304 .end
00000000'GF 03 00000000'8F 03 0171 305
00000000'GF 03 00000000'8F 03 0171 306
00000000'GF 03 00000000'8F 03 0171 307
00000000'GF 03 00000000'8F 03 0171 308
00000000'GF 03 00000000'8F 03 0171 309
00000000'GF 03 00000000'8F 03 0171 310
00000000'GF 03 00000000'8F 03 0171 311
00000000'GF 03 00000000'8F 03 0171 312
00000000'GF 03 00000000'8F 03 0171 313
00000000'GF 03 00000000'8F 03 0171 314
00000000'GF 03 00000000'8F 03 0171 315
00000000'GF 03 00000000'8F 03 0171 316
00000000'GF 03 00000000'8F 03 0171 317
00000000'GF 03 00000000'8F 03 0171 318
00000000'GF 03 00000000'8F 03 0171 319
00000000'GF 03 00000000'8F 03 0171 320
00000000'GF 03 00000000'8F 03 0171 321
00000000'GF 03 00000000'8F 03 0171 322
00000000'GF 03 00000000'8F 03 0171 323
00000000'GF 03 00000000'8F 03 0171 324
00000000'GF 03 00000000'8F 03 0171 325
00000000'GF 03 00000000'8F 03 0171 326
00000000'GF 03 00000000'8F 03 0171 327
00000000'GF 03 00000000'8F 03 0171 328
00000000'GF 03 00000000'8F 03 0171 329
00000000'GF 03 00000000'8F 03 0171 330
00000000'GF 03 00000000'8F 03 0171 331
00000000'GF 03 00000000'8F 03 0171 332
00000000'GF 03 00000000'8F 03 0171 333
00000000'GF 03 00000000'8F 03 0171 334
00000000'GF 03 00000000'8F 03 0171 335
00000000'GF 03 00000000'8F 03 0171 336
00000000'GF 03 00000000'8F 03 0171 337
00000000'GF 03 00000000'8F 03 0171 338
00000000'GF 03 00000000'8F 03 0171 339
00000000'GF 03 00000000'8F 03 0171 340

PLISPUTLISTITEM
Symbol table

ATR_M_RECUR	=	00000008		
ATR_V_PRINT	=	00000007		
FAIC		00000158	R	02
FCB_B_ENVIR		000001C2		
FCB_B_ESA		0000012E		
FCB_B_EXTRA		0000003D		
FCB_B_FAB		000000A6		
FCB_B_IDENT		00000040		
FCB_B_IDENT_NAM		00000042		
FCB_B_NAM		000000F6		
FCB_B_NUMKCBS		0000003C		
FCB_B_RAB		00000062		
FCB_C_LEN		000001C2		
FCB_C_STRLLEN		00000034		
FCB_L_ATTR		0000000C		
FCB_L_BUF		00000014		
FCB_L_BUF_END		00000018		
FCB_L_BUF_PT		0000001C		
FCB_L_CNDADDR		000001B2		
FCB_L_CONDIT		000001AE		
FCB_L_DTR		00000010		
FCB_L_ERROR		00000008		
FCB_L_KCB		00000038		
FCB_L_NEXT		00000000		
FCB_L_PREVIOUS		00000004		
FCB_L_PRN		00000034		
FCB_Q_RFA		00000020		
FCB_W_COLUMN		0000002E		
FCB_W_IDENT_LEN		00000040		
FCB_W_LINE		00000030		
FCB_W_LINESIZE		0000002A		
FCB_W_PAGE		00000032		
FCB_W_PAGESIZE		0000002C		
FCB_W_REVISION		00000028		
GETOPT_B_BITS		00000009		
GETOPT_B_TMO		00000008		
GETOPT_C_LEN		0000000A		
GETOPT_L_FXDCTL		00000000		
GETOPT_L_PROMPT		00000004		
PLISPUTNLIS_R6	*****		X	02
PLISBITCHAR_R6	*****		X	02
PLISFIXBVCHA_R6	*****		X	02
PLISFLXBVCHA_R6	*****		X	02
PLISFLTBVCHA_R6	*****		X	02
PLISFLTDVCHA_R6	*****		X	02
PLISIO_ERROR	*****		X	02
PLISPICVCHA_R6	*****		X	02
PLISPUTLBIT_R6	000000AE	RG		02
PLISPUTLBIT_R6	00000068	RG		02
PLISPUTLCHAR_R6	00000000	RG		02
PLISPUTLFIXB_R6	00000086	RG		02
PLISPUTLFIXD_R6	000000D7	RG		02
PLISPUTLFLTB_R6	000000F8	RG		02
PLISPUTLFLTD_R6	00000119	RG		02
PLISPUTLPIC_R6	0000013A	RG		02
PLISPUTLVCHA_R6	00000061	RG		02
PLIS_ERROR	*****		X	02

F 13

16-SEP-1984 02:24:21 VAX/VMS Macro V04-00
6-SEP-1984 11:39:33 [PLIRTL.SRC]PLISPUTLIS.MAR;1

Page 7
(1)

PLIS_STROVFL	*****	X	02
SIZ...	=	00000001	
STK_L_AP		00000008	
STK_L_ARG_LIST		FFFFFFFFFF8	
STK_L_CND_HND		00000000	
STK_L_CND_LST		FFFFFFFFFF4	
STK_L_DISPLAY		FFFFFFFFFFC	
STK_L_FP		0000000C	
STK_L_PC		00000010	
STK_L_PSL		00000004	
STK_L_REGS		00000014	
STR_B_FIELD		00000018	
STR_C_LEN		00000008	
STR_L_FLD_END		00000014	
STR_L_FLD_PT		00000010	
STR_L_FP		00000004	
STR_L_FS		0000000C	
STR_L_PARENT		00000008	
STR_L_SP		00000000	
STR_L_STACK		00000004	
STR_L_STACK_END		00000408	

PL
SY
RM
SI
SY
SY
PS
--
SA
-F
PH
--
Ir
Cc
Pa
Sy
Pa
Sy
Ps
Cr
As
Th
44
Th
43
23
Ma
--
-9
-9
TC
8C
TF
MA

```
+-----+
! Psect synopsis !
+-----+
```

PSECT name	Allocation	PSECT No.	Attributes
ABS .	00000000 (0.) 00 (0.) NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE		
\$ABSS	FFFFFFFC (0.) 01 (1.) NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE		
_PLISCODE	00000171 (369.) 02 (2.) PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG		

```
+-----+
! Performance indicators !
+-----+
```

Phase	Page faults	CPU Time	Elapsed Time
Initialization	12	00:00:00.08	00:00:00.29
Command processing	74	00:00:00.62	00:00:01.95
Pass 1	251	00:00:09.47	00:00:19.42
Symbol table sort	0	00:00:01.34	00:00:02.66
Pass 2	59	00:00:01.62	00:00:03.81
Symbol table output	10	00:00:00.08	00:00:00.28
Psect synopsis output	1	00:00:00.03	00:00:00.23
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	407	00:00:13.24	00:00:28.65

The working set limit was 1050 pages.

51576 bytes (101 pages) of virtual memory were used to buffer the intermediate code.

There were 50 pages of symbol table space allocated to hold 972 non-local and 8 local symbols.

304 source lines were read in Pass 1, producing 11 object records in Pass 2.

18 pages of virtual memory were used to define 16 macros.

```
+-----+
! Macro library statistics !
+-----+
```

Macro library name	Macros defined
\$255\$DUA28:[PLIRTL.OBJ]PLIRTMAC.MLB;1	6
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	7
TOTALS (all libraries)	13

995 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=TRACEBACK/LIS=LI\$\$.PLIPUTLIS/OBJ=OBJ\$\$.PLIPUTLIS MSRC\$\$.PLIPUTLIS/UPDATE=(ENH\$\$.PLIPUTLIS)+LIB\$\$.PLIRTM

0308 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PLIFORMAT
LIS

PLIGETBUF
LIS

PLIGETEDI
LIS

PLIHEEP
LIS

PLIMSGTXT
LIS

PLIPUTFIL
LIS

PLIRMSBIS
LIS

PLIRECPT
LIS

PLIREAD
LIS

PLIREWRT
LIS

PLIOPEN
LIS

PLIPROTEC
LIS

PLIPUTEDI
LIS

PLIGETLIS
LIS

PLIPKDIVL
LIS

PLIPUTLIS
LIS

PLIMSGPTR
LIS

PLIPKDIVS
LIS

PLIPUTBUF
LIS

PLIGETFIL
LIS